Systèmes Multi-Agents

FIPA - Jade

BATTACH Marwene

Université de Sousse - ISET de Sousse

Mastère M1 GLDRA

- Protocoles de communication
  - Protocole Subscribe
  - Protocole AchieveRE
  - Protocole AchieveRE Simplifié
- Cas d'étude
- Cas d'étude - Solution sans protocole
- Cas d'étude - Solution avec protocole

Généralités sur FIPA (Foundation of Intelligent Physical Agents)

Architecture abstraite

- But : favoriser l'interopérabilité et la réutilisabilité
- Identifier les entités architecturales abstraites et leurs relations
  - Transport de messages, langage de communication, services de renseignements, langages de contenu
- Identifier les entités difficilement définissable de façon abstraite
  - Gestion et mobilité des agents

Transports de messages

- But : gérer la livraison et la représentation des messages, des protocoles de communications inter-réseaux et inter-plateformes
  - Message = enveloppe + corps
  - 1 MTS sur chaque plateforme

Généralités sur FIPA (Foundation of Intelligent Physical Agents)

Gestion des agents

- But : fournir la structure permettant l'existence et le fonctionnement des agents
- Modèle de référence logique pour la création, l'enregistrement, la localisation, la communication, la migration et le retrait d'agents
- Décrit les primitives et ontologies nécessaires pour les services
  - *Pages Blanches* : localisation, désignation et contrôle de l'accès aux services (AMS)
  - *Pages Jaunes* : localisation, et enregistrement des services (DF)
  - Service de transport de messages

Généralités sur FIPA (Foundation of Intelligent Physical Agents)

Communication entre agents

- Techniques de communication pour structurer les interactions dans le système
- Spécifications du langage de communication + protocoles d'interaction + bibliothèques d'actes communicatifs prédéfinis + protocoles d'interaction + langages de contenu

Plateforme FIPA

- Plateformes dites "FIPA-compliant" :
  - JADE (Java Agent DEvelopment)
  - JACK (Agent Oriented Software Group)

Généralités sur JADE

JADE : Java Agent DEvelopment framework

- But : développement et exécution de systèmes multi-agent conformes aux normes FIPA
  - service de nommage
  - service de pages jaunes
  - transport de messages
  - service d'analyse
  - bibliothèque des protocoles d'interaction de FIPA
- Les agents sont des coquilles auxquelles il faut ajouter des comportements implémentant des services/fonctionnalités
- Les communications utilisent le standard ACL
- Possibilité de communications entre plateformes JADE

Généralités sur JADE

Plateforme

- Comprend 3 composantes de base :
  - un environnement d'exécution dans lequel les agents "existent"
  - une librairie de classes java utilisable directement ou par extension pour développer les agents
  - une suite d'outils graphiques permettant l'administration et la gestion des agents actifs

Caractéristiques de JADE

- Entièrement implémentée en JAVA
- Conforme aux spécifications FIPA
- Open Source et distribué avec licence LGPL
- Distribution possible sur différents serveurs
- Modifiable en cours d'exécution (mobilité des agents)

Architecture de JADE

Architecture

- Chaque instance de JADE est un Conteneur (Container)
- Plateforme = ensemble de Conteneurs
  - obligation d'avoir un Conteneur Principal (Main Container) actif
  - d'autres conteneurs peuvent s'y enregistrer
- Le conteneur principal possède 2 agents spéciaux
  - AMS (Agent Management System) : Système de gestion d'agents
  - DF (Directory Facilitator) : Service de pages jaunes

Architecture de JADE

Conteneur

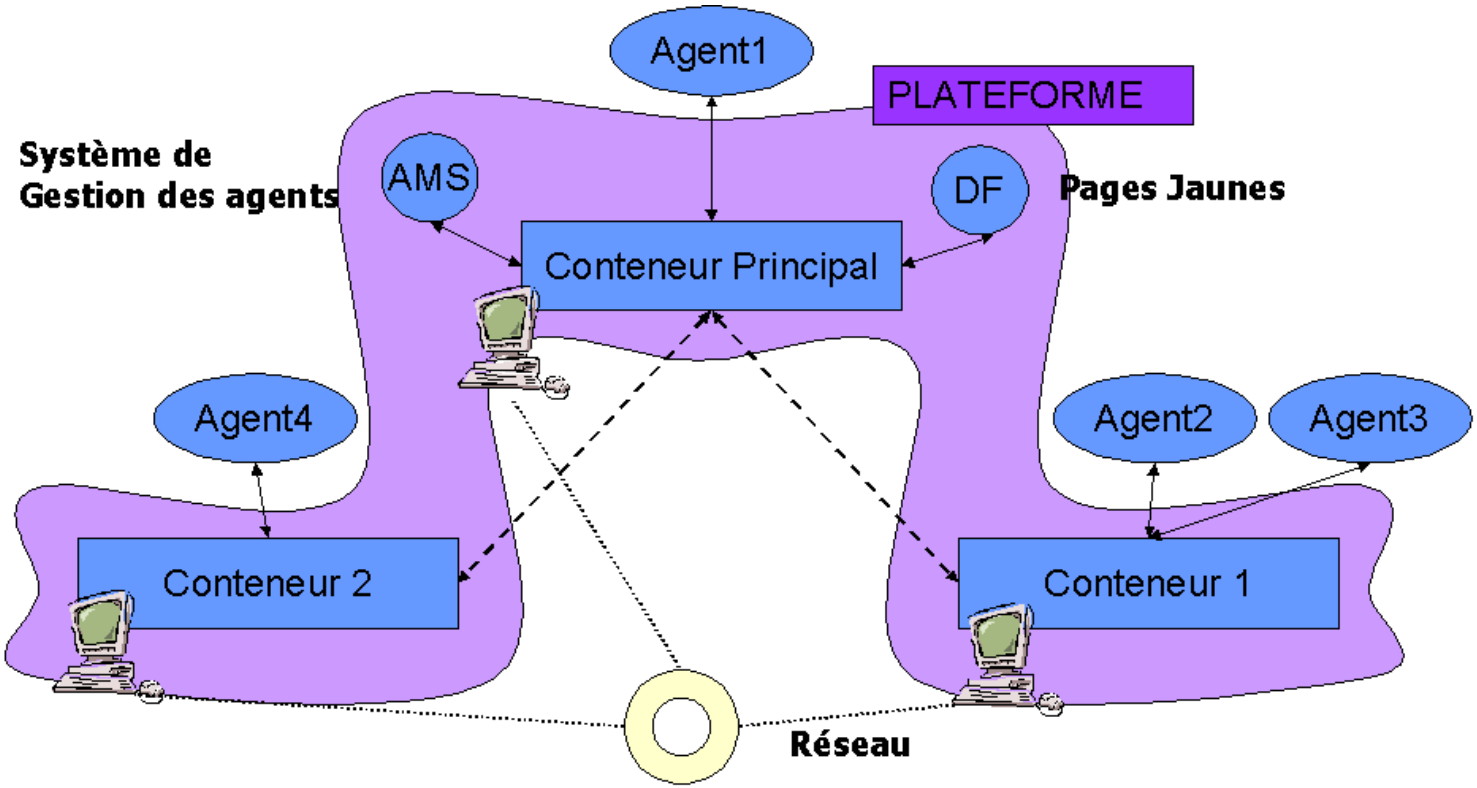
Chaque conteneur d'agents :

- environnement multi-threads composé d'un thread d'exécution pour chaque agent
- gère localement un ensemble d'agents
- règle le cycle de vie des agents (création, attente et destruction)
- assure le traitement des communications :
  - répartition des messages ACL reçus
  - routage des messages
  - dépôt des messages dans les boîtes privées de chaque agent
  - gestion des messages vers l'extérieur

Architecture de JADE

Exemple d'architecture

UNE plateforme avec 3 conteneurs et 4 agents

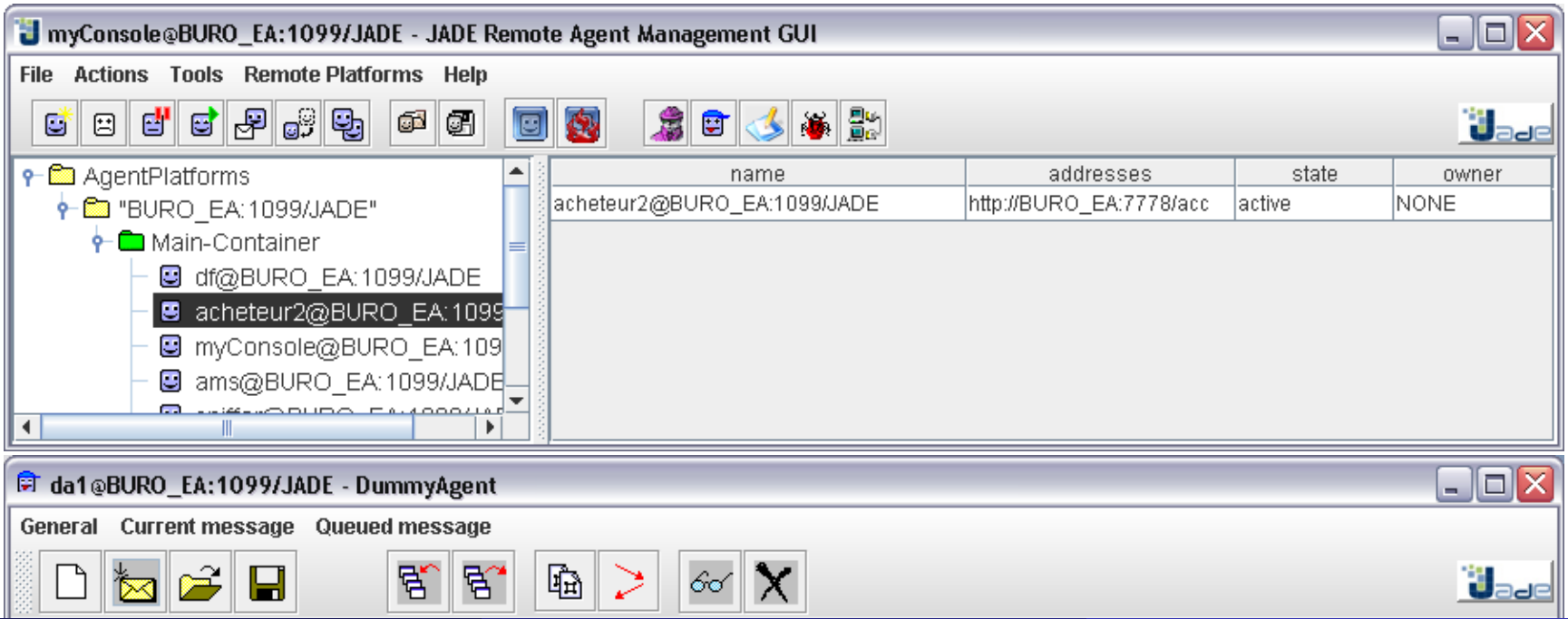


Outils de JADE

Outils principaux

JADE propose quelques outils de gestion dont :

- JADE RMA (Remote Agent Management) : gestion des agents d'une plateforme
- JADE Dummy Agent : permet d'envoyer des messages aux agents
- JADE Sniffer : analyse des messages échangés entre agents
- JADE Introspector : affiche le détail du cycle de vie d'un agent



Agents en JADE

Agent JADE

- coquille à laquelle il faut ajouter des comportements implémentant des services/fonctionnalités
- Suit son cycle de vie

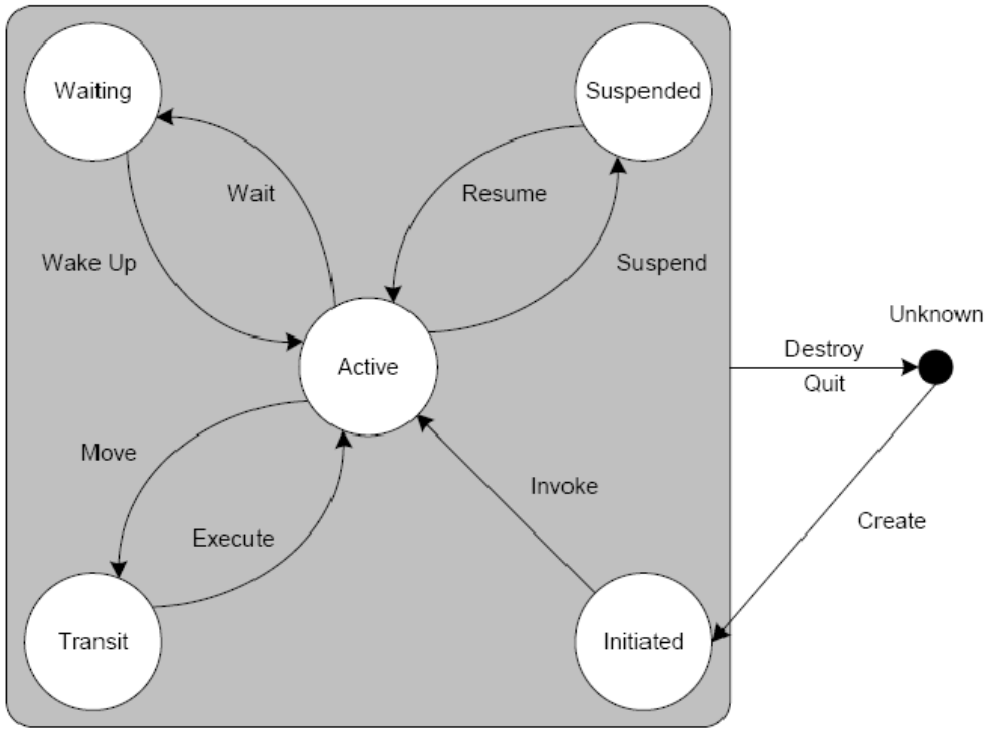


Figure 3 - Agent life-cycle as defined by FIPA.

Agents en JADE

Agent JADE - Elément de programmation

- hérite de la classe Agent
- Thread
- Possibilité de déclarer/rechercher un *service*
  - créer des groupes d'agents répondant à une demande
  - facilite la recherche d'agents répondant à un besoin
  - Gérés par Pages Jaunes
  - Proche de la notion de Web Services
- Méthode *setup*() invoquée dès la création de l'agent
- Méthode *takedown*() invoquée avant qu'un agent ne quitte la plateforme (soit détruit)

Exemple de l'agent HelloWorld I

```
import jade.core.Agent;
import static java.lang.System.out;

public class AgentHello extends Agent {
 private String texteHello;
 /** Initialisation de l'agent */
 protected void setup() {
  texteHello = "Bonjour a toutes et a tous";
  out.println("De l'agent "+ getLocalName() + " : " + texteHello);
  out.println("Mon adresse est "+ getAID());
  //tuer l'agent
  doDelete();
  }
 }
 /** Adieu */
 protected void takeDown() {
  out.println("Agent " + getLocalName() + " part.");
 }
}
```

Exécuter des agents HelloWorld

Lancer la plateforme et les agents

- Il faut premièrement demander le chargement de la plateforme
- puis demander le chargement des agents :
  `java jade.Boot -gui a1:AgentHello;a2:AgentHello`

Résultat de l'exécution

```
oct. 01, 2018 4:30:38 PM jade.core.Runtime beginContainer
...
Agent container Main-Container@10.4.151.26 is ready.
...
De l'agent a1 : Bonjour à toutes et à tous
De l'agent a2 : Bonjour à toutes et à tous
Mon adresse est ( agent-identifier :name a2@10.4.151.26:1099/JADE  :address
Agent a2 quitte la plateforme.
Mon adresse est ( agent-identifier :name a1@10.4.151.26:1099/JADE  :address
Agent a1 quitte la plateforme.
```